



Uso de la interface Set

Transcripción

[00:00] ¿Cómo vamos? Vamos a iniciar ahora nuestra clase 12 dentro del bloque usando la interface set. Ahora vamos a ver el uso de la interface set, cómo usarla y por qué usarla. Vamos a hacer la clase 11, vamos a duplicarla, vamos a colocar clase 12 y vamos a dar un pequeño overview. Nuestro pequeño overview va a ser el siguiente. Vamos a hacer un diseño simple utilizando el Draw.io y aquí estuve creando lo siguiente.

[00:36] El set aquí tenemos por ejemplo toda la lista de nuestro set y dentro de ella tenemos nuestros objetos. Que por ejemplo uno es Luis Miguel, otro es Pedrito los palotes, otro es Gustavo Sánchez, otro es Marcia María y etcétera, no tiene una posición fija, esa es una desventaja por ejemplo, comparado con la interface list, que tenemos una lista, que tenemos secuencial, podemos saber aquí su posición 0, 1, 2, 3 y 4 y ahí podemos obtenerlas.

[01:08] Entonces esa es una diferencia de ellas. Otra es que no podemos tener objetos repetidos, por ejemplo, vamos a ver lo siguiente. Imaginémonos que yo hago dos veces Marcia María y coloco aquí también dos veces Gustavo Sánchez. Si fuese una interface list, ¿qué sucedería? Listaría los ocho, ¿no?

[01:46] Pero en el set prácticamente él no va a colocar objetos duplicados de string, que sería aquí Juan Carlos, Pedro Pedrito, Gustavo Sanchez, Pepito los palotes, Marcia Maria, Luis Miguel.

[01:36] Solamente aparecen una vez, es una diferencia también que podemos evitar objetos duplicados, y es muy importante también poder tener esa

opción. Otra desventaja por ejemplo, una desventaja podría ser, ¿qué cosa? Tenemos aquí `listaAlumnos.get`. No existe el `get` en la posición 1, 2, 0 como había mencionado anteriormente. Pero podemos recorrer la lista.

[02:30] Aquí tenemos una lista de alumnos y podemos ver aquí cuáles son los alumnos también podemos utilizar `igual con`, utilizar `lambda .forEach`. Vamos a colocar aquí `alumno`, vamos a colocar aquí `System.out.println`, y aquí por ejemplo ponemos aquí `alumno`. Prácticamente aquí tendríamos el mismo resultado.

[03:18] ¿Por qué sería una ventaja? Porque imaginémonos ya no puedo registrar dos alumnos iguales, esa es una ventaja de utilizar el `set` por ejemplo para esa situación o una circunstancia inespecífica.

[03:28] O por ejemplo, número de tarjeta. No puedo colocar dos números de tarjetas duplicadas, cosa que en un `list` podría permitir. Una desventaja es que aquí por ejemplo no podemos obtener Gustavo Sánchez. ¿Podemos buscarlo? Sí. Recorriendo uno por uno y ver cuál es el que tiene Gustavo Sánchez. Esto podría ser una opción.

[03:52] Pero a su vez también por ejemplo podemos tener la situación, imaginémonos, no sabemos qué cosa queremos trabajar y como todo viene de `collection`, de acuerdo a nuestra documentación aquí, todo viene aquí de `collection`, entonces podemos hacer lo siguiente.

[04:10] Vamos a colocar aquí `Collection` y vamos a importar. Entonces en un inicio no sabemos qué cosa queremos. Quiero un `HashSet`. Todo bien. Ahora no, quiero un `ArrayList`. Vamos a ver aquí `ArrayList`. Perfecto, `ArrayList`. Muy bien, ya registró. Aquí también vamos a ver lo siguiente. Vamos a colocar aquí `alumno7` y `alumno8`. Ya que estamos haciendo aquí, vamos a verificar nuevamente.

[04:52] Imprimió dos veces Marcia María, imprimió dos veces Gustavo Sánchez. Y prácticamente en el orden que yo coloqué, porque está en un

ArrayList. Ahora colocamos HashSet. El orden es lo de menos, pero solamente imprimió una vez Marcia María y una vez el alumno Gustavo Sánchez.

[05:17] Por eso es muy bueno siempre tener la parte de la documentación de Java. Es bien importante porque así podemos saber cómo podemos jugar con nuestras interfaces, con nuestras clases, porque aquí ya sabemos, estamos utilizando la clase pai, la clase padre, que por ejemplo ahí ya podemos heredar.

[05:34] Tenemos el set, el list. ¿Puedo utilizar el list o puedo utilizar set? ¿Cómo? Trabajando con collection desde el inicio. Así, si al inicio no sabemos qué cosa deseamos trabajar, primero utilizamos, declaramos con collection y vamos utilizando el HashSet, después podemos utilizar el ArrayList, LinkedList y ver con cuál forma se vería mejor para una situación inespecífica, que es lo que estamos necesitando.

[06:05] Esto básicamente sería todo en nuestra clase 12. Espero que haya sido claro la parte de diferencias entre uno, entre otro. También ver en la parte de jerarquías, que es muy importante, y ver que aquí, si hago una lista, él va a recorrer primero aquí, primero aquí, primero aquí o después aquí, después aquí, en el orden que el for, en el orden que él encuentre a los objetos. Entonces esto sería todo. Muchas gracias y nos vemos en la siguiente clase.